结合用户兴趣度聚类的协同过滤推荐算法 *

黄贤英，龙姝言†，谢　晋

(重庆理工大学 计算机科学与工程学院，重庆 400054)

摘　要：针对传统的协同过滤算法忽略了用户兴趣源于关键词以及数据稀疏的问题，提出了结合用户兴趣度聚类的协同过滤推荐算法。利用用户对项目的评分，并从项目属性中提取关键词，提出了一种新的 RF-IIF (rating frequency- inverse item frequency)算法，根据目标用户对某关键词的评分频率和该关键词被所有用户的评分频率，得到用户对关键词的偏好，形成用户—关键词偏好矩阵，并在该矩阵基础上进行聚类。然后利用 logistic 函数得到用户对项目的兴趣度，明确用户爱好，在类簇中寻找目标用户的相似用户，提取邻居爱好的前 N 个物品对用户进行推荐。实验结果表明，算法准确率始终优于传统算法，对用户爱好判断较为准确，缓解了数据稀疏问题，有效提高了推荐的准确率和效率。

关键词：协同过滤；推荐算法；用户兴趣；K-means 聚类

中图分类号：TP391　　doi: 10.3969/j.issn.1001-3695.2018.03.0149

Collaborative filtering recommendation algorithm combined with user interest degree clustering

Huang Xianying, Long Shuyan†, Xie Jin

(*School of Computer Science & Engineering*, *Chongqing University of Technology*, *Chongqing* 400054, *China*)

Abstract: Aiming at the problem of ignores the user's interest in the key words and the data sparseness in traditional collaborative filtering algorithm. We proposed a collaborative filtering recommendation algorithm combined with the user interest degree clustering. We using user ratings for projects and extracting keywords from item attributes. A new Rating Frequency-Inverse Item Frequency algorithm is proposed. According to the target users' scoring frequency for a key word and the frequency of the keyword being evaluated by all users. We get users' preferences for keywords, form user preference matrix, and cluster on the basis of this matrix. Then we use logistic function to get users' interest in projects. Clear user preferences and find similar users of target users in the clusters. Then extract N items from neighbors' preferences, and recommend users. Experimental results show that the algorithm accuracy rate is always better than the traditional algorithm. it, s more accurate to judge the user interest, alleviating the problem of data sparseness, and effectively improves the accuracy and efficiency of recommendation.

Key words: Collaborative filtering; recommendation algorithm; user interest; K-means clustering

0　引言

推荐系统[1](recommender system，RS)是为用户推荐有用的项目的一种软件工具，也可以说是一种技术方法。早期的推荐系统为用户提供的推荐都是当前流行且大众化的内容，并不能满足个体用户的需求，因此产生了基于个性化推荐系统。这类个性化推荐最简单的方法就是根据用户的历史行为数据（包括评分，历史记录）得到个性化需求，来预测可能的最适项目。根据用户需求的不同，推荐系统也出现了多种方法，常用的有基于内容的推荐方法，基于用户的推荐方法、基于组合的推荐方法、基于关联规则的推荐方法和基于协同过滤的推荐方法[2]。其中协同过滤推荐算法是当前各类推荐算法中研究最多且推荐效果最好的算法之一。因此，协同过滤推荐算法也广泛应用于各大电子商务网站中，例如淘宝、京东商城、亚马逊、当当网等。

协同过滤算法又分为基于模型的协同过滤推荐和基于内存的协同过滤推荐[3-5]。其中基于内存的协同过滤推荐算法又分为基于用户的协同过滤推荐算法和基于项目的协同过滤推荐算法。基于用户的协同过滤推荐通过计算目标用户与其余用户之间的相似度，得到与目标用户兴趣爱好相似的用户，根据相似用户

收稿日期：2018-03-02；**修回日期：**2018-04-11　**基金项目：**国家社会科学基金资助项目（17XXW004）；国家自然科学基金资助项目（61603065）；国家统计局全国统计科学研究重点项目（2016LZ08）；国家教育部人文社会科学研究项目（15YJC790061）

作者简介：黄贤英（1967-），女，重庆万州人，教授，硕士，主要研究方向为数据挖掘、推荐系统、嵌入式系统、信息检索；龙姝言（1992-），女，重庆江津人，硕士研究生，主要研究方向为推荐系统、数据挖掘（lsy0727@foxmail.com）；谢晋（1993-），男，湖北十堰人，硕士研究生，主要研究方向为自然语言处理、信息检索.

的喜好为目标用户进行推荐。随着用户和项目数量的增加，数据稀疏[6]、相似度计算不准确和实时性差成为影响推荐系统性能的关键因素。

为了解决数据稀疏导致的评分预测问题，相关学者引入聚类算法进行优化研究，例如文献[7]提出了一种结合项目聚类和 Slope one 方案的推荐算法，利用项目聚类算法，将项目聚合为几个类簇，并且将 Slope one 算法应用到每个类簇中，对目标用户对未知项目的评分进行预测。文献[8]提出了一种通过矩阵聚类的协同过滤算法，对用户评分数据利用矩阵聚类算法进行聚类，然后将协同过滤算法应用到聚类后的子矩阵上，该方法提高了算法的推荐精度。文献[9]等提出了基于项目聚类的协同过滤推荐算法，通过分析用户对项目的评分，来对项目进行聚类，再在相似类簇中搜索目标项目的最近邻，该算法有效提高了推荐系统的实时响应速度。上述改进算法虽然在一定程度上缓解了传统协同过滤推荐算法的问题，但依然忽略了用户对项目的兴趣来源于关键词，用户对项目的某一关键词感兴趣，所以才会产生对该项目的评分。

针对上述问题，本文提出了一个以用户与-关键词关系为中心的协同过滤算法，将用户-项目矩阵与项目-关键词矩阵结合在一起，构成用户-关键词矩阵，再引入 $RF-IIF$ 算法计算用户对关键词的偏好，形成用户偏好向量，对用户进行聚类，再利用 logistic 函数计算用户对项目的兴趣度，进而在类簇中寻找目标用户的相似邻居，使算法更加高效。该算法使用结合用户兴趣度的聚类算法，使系统能够充分了解用户和项目，发现用户之间的隐藏关系，使一些冷门项目可能被推荐，并提高了算法的实时性。

1 协同过滤推荐算法简介

协同过滤算法的基本思想概括起来就是为用户推荐兴趣爱好类似的用户感兴趣的项目[10]。算法分为 4 个步骤：首先，收集用户评分数据，构建用户-项目评分矩阵，利用评分矩阵对用户进行相似度的计算，然后根据相似度计算的结果选取 top-N 个最近邻，最后根据这些最近邻的评分数据计算预测目标用户的评分，依据预测评分得到推荐结果。协同过滤推荐算法不依赖用户和项目本身的特征与属性，而是分析更能够反应用户偏好的历史数据，找到用户之间的隐藏关系，使得推荐结果更加贴近用户的喜好。

1.1 评分矩阵

定义 1 用户-项目矩阵。用户的所有评分记录可以看作是一个用户-项目评分矩阵 R，其中包括 m 个用户 $U=\{u_1,u_2,u_3,\cdots,u_m\}$ 以及 n 个项目 $\{I_1,I_2,I_3,\cdots,I_n\}$ 如式(1)所示。本文采用 r_{ij} 表示用户 u_i 对项目 I_j 的评分。分值为 1~5 分，分数越高，表示该用户对该物品的喜欢程度越深。

$$R=\begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_m \end{array}\begin{array}{c} \begin{array}{ccccc} I_1 & I_2 & I_3 & \cdots & I_n \end{array} \\ \begin{bmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1n} \\ r_{21} & r_{22} & r_{23} & \cdots & r_{2n} \\ r_{31} & r_{32} & r_{33} & \cdots & r_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & r_{m3} & \cdots & r_{mn} \end{bmatrix} \end{array} \tag{1}$$

1.2 相似度计算

协同过滤算法的目的是对目标用户 u 的未评分项目进行预测评分。为了预测评分，首先就需要找到用户的相似邻居集，用户间相似性的计算就成了协同过滤算法的关键之一。常用的相似性度量方法有余弦相似度[11]、Pearson 相关系数[12]和修正的余弦相似度。由于余弦相似度忽略了不同用户之间的不同评分标准，学者们又提出了修正的余弦相似度计算方法，将用户评分减去该用户的平均评分之后，再进行相似度的计算，计算方法如下：

$$sim(i,j)=\frac{\sum_{u\in U_{ij}}(r_{u,i}-\bar{r}_u)(r_{u,j}-\bar{r}_u)}{\sqrt{\sum_{u\in U_{ij}}(r_{u,i}-\bar{r}_u)^2}\sqrt{\sum_{u\in U_{ij}}(r_{u,j}-\bar{r}_u)^2}} \tag{2}$$

其中：将评分矩阵 R 看做向量空间，u 、v 表示两个不同的用户，i 、j 表示不同的项目，$r_{u,i}$ 表示用户 u 对项目 i 的已有评分，$\vec{i}$ 为所有用户对项目 i 的评分向量，$\vec{j}$ 为所有用户对项目 j 的评分向量，U_{ij} 表示对 i 和 j 都评分的用户集合，$\bar{r}_u$ 表示用户 u 对他所评分项目的平均值。

1.3 评分预测

通过相似度计算得到目标用户的最近邻居集 NN，通过式(3)预测用户 u 对项目 I_i 的评分 $P_{u,i}$。

$$P_{u,i}=\bar{r}_i+\frac{\sum_{j\in NN}sim(i,j)(r_{u,j}-\bar{r}_j)}{\sum_{j\in NN}|sim(i,j)|} \tag{3}$$

其中：$\bar{r}_i$ 表示用户对项目 i 的平均评分，$\bar{r}_j$ 表示用户对项目 j 的平均评分，$sim(i,j)$ 表示项目 i 和 j 之间的相似度。

根据上述方法预测目标用户 u 对未评分项目的评分，并选取评分的前 N 个项目，将其推荐给目标用户 u 。

2 结合用户兴趣度聚类的协同过滤推荐算法

本文提出的结合用户兴趣度聚类的协同过滤推荐算法，根据用户对项目的评分记录和项目关键词，得到用户对关键词的评分情况；然后采用 $RF-IIF$ 算法得到用户对不同关键词的偏好程度，在该偏好程度矩阵的基础上，对用户进行聚类。再结合 Logistic 函数得到用户对项目兴趣度矩阵；最后，利用该矩阵在聚类得到的相似用户类簇中寻找目标用户的最近邻居。算法具体流程如图 1 所示。

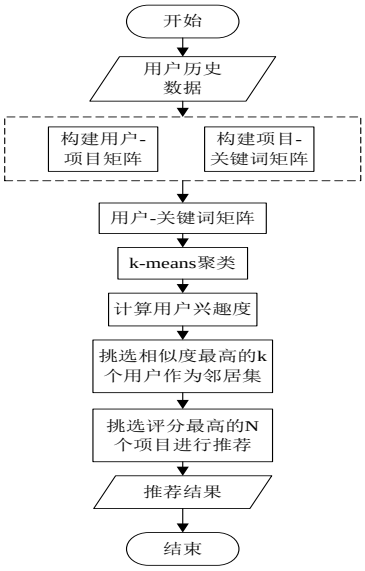

图 1 传统的协同过滤推荐算法流程图

2.1 用户-关键词偏好度矩阵计算

不同用户选择某个项目通常是由于对该项目的某个关键词属性感兴趣，而用户对项目每个关键词的兴趣度也不是同一而论的。因此，可以将用户对项目的评分映射到项目相应的关键词属性上，有些没有相同评分项目的用户也能借助对某些关键词属性的评分而进行相似性的度量。用户对关键词属性的评分能够在一定程度上体现用户的偏好，因此，本文通过用户-项目评分次数矩阵和项目属性矩阵计算得到用户-关键词矩阵。

定义 2 项目—关键词矩阵. 在推荐系统中，项目都具有若干属性来描述项目本身，将系统中项目的关键词表示为 $T=\{t_1,t_2,\cdots,t_r\}$，项目—关键词矩阵如式(4)所示。

$$K=\begin{bmatrix}1, & 0, & \cdots, & 0\\ 1, & 1, & \cdots, & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0, & 1, & \cdots, & 1\end{bmatrix} \tag{4}$$

其中：“1”表示某个项目具有某个关键词属性，“0”表示某个项目不具有某个关键词属性，矩阵中的任意列为向量 T_q。在项目属性列表中，每个项目都有项目 ID，发布时间，项目类型等关键属性值，可以从这些属性值中提取出每个项目对应的属性值关键词。将用户对项目的评分映射到项目的属性值上，生成用户-关键词评分矩阵。

定义 3 用户-关键词评分矩阵. 由用户-项目评分矩阵 R 和项目-关键词矩阵 K，得到用户-关键词评分矩阵 W，具体公式如式(5)所示。

$$\begin{aligned}W=R\bullet K&=\begin{bmatrix}R_1\\ R_2\\ \vdots\\ R_m\end{bmatrix}\bullet\left[T_1,T_2,\cdots,T_r\right]\\ &=\begin{bmatrix}R_1\bullet T_1 & R_1\bullet T_2 & \cdots & R_1\bullet T_r\\ R_2\bullet T_1 & R_2\bullet T_2 & \cdots & R_2\bullet T_r\\ \vdots & \vdots & \ddots & \vdots\\ R_n\bullet T_1 & R_n\bullet T_2 & \cdots & R_n\bullet T_r\end{bmatrix}\end{aligned} \tag{5}$$

在矩阵 W 中，用户对不同的关键词有不同的评分，即用户对不同关键词的偏好不同，即用户 u_i 对关键词 t_j 的偏好程度随用户 u_i 对 t_j 的评分次数增加而增加，随着关键词的流行度增加而下降。由此特性，本文提出一种 $RF-IIF$ 算法根据用户对关键词的评分次数来预测用户对关键词的偏好，并根据偏好来对用户进行聚类。

评分频率（RF）表示目标用户对某关键词的评分次数，计算方法如式(6)所示。

$$RF_{i,j}=\frac{w_{i,j}}{\sum_{q\in T}w_{i,q}} \tag{6}$$

其中：$w_{i,j}$ 为矩阵 W 中用户 u_i 对关键词 t_j 的评分次数，T 为矩阵中全部关键词的集合，$w_{i,q}$ 为用户 u_i 评分次数总合。

反向项目频率（IIF）表示评分该关键词的用户总数，计算方法如式(7)所示。

$$IIF_j=\lg\frac{n}{N_j} \tag{7}$$

其中：n 为矩阵 W 中的用户个数总数，N_j 为矩阵中评分过关键词 t_j 的用户个数。

那么用户 u_i 对关键词 t_j 的偏好程度可以由式(8)所示。

$$Pre=RF_{i,j}\times IIF_j=\frac{w_{i,j}}{\sum_{q\in T}w_{i,q}}\times\lg\frac{n}{N_j} \tag{8}$$

用户 u_i 对于所有关键词 $T=\{t_1,t_2,\cdots,t_r\}$ 的偏好程度组成用户 u_i 的偏好向量 $Pre_i=(Pre_{i,1},Pre_{i,2},\cdots,Pre_{i,r})$，所有用户的偏好向量就构成了用户-关键词偏好矩阵如式(9)所示。

$$Pre=\begin{bmatrix}Pre_{1,1} & Pre_{1,2} & \cdots & Pre_{1,r}\\ Pre_{2,1} & Pre_{2,2} & \cdots & Pre_{2,r}\\ \vdots & \vdots & \ddots & \vdots\\ Pre_{m,1} & Pre_{m,2} & \cdots & Pre_{m,r}\end{bmatrix} \tag{9}$$

由 $RF-IIF$ 算法可以得出，对于用户评分多的关键词，即热门关键词，根据式(8)计算得出的偏好值偏低。而对于那种冷门的关键词，用户如果对其评分，表示这个关键词对于该用户相对于其他用户的重要程度更高，更受该用户关注，这样就能更好的区分用户偏好。

2.2 用户聚类查找最近邻

在上节中，利用用户—项目评分矩阵和项目—关键词矩阵得到了用户偏好度矩阵，在一定程度上明确了用户偏好。但在实际推荐系统中，得到的用户—关键词偏好矩阵依然是个高维稀疏矩阵。因此，采用 K-means 算法对用户进行聚类，将用户划分成为规模较小的相似类簇，在类簇中寻找相似邻居，缓解数据稀疏问题。

K-means 聚类通过计算对象之间的距离来衡量他们的相似度，将小于距离阈值的对象作为相似类簇。由于余弦相似度更适用于数据稀疏性较强的情况，本文采用余弦距离来计算相似度进行聚类：

$$S(u_a,u_b)=\frac{Pre_{u_a}\bullet Pre_{u_b}}{\|Pre_{u_a}\|\bullet\|Pre_{u_b}\|} \tag{10}$$

其中：Pre_a 和 Pre_b 分别表示用户 u_a 和 u_b 的偏好向量。采用 K-means 聚类算法将具有相同关键词偏好的用户划分为一个类簇。

关键词评分次数表示了这个用户对该关键词的喜好程度，通常来说，用户越喜欢该关键词，评分次数越多。相对于通过评分来考虑用户偏好，评分次数更能够准确判断用户兴趣。由于用户对不同关键词的评分次数差别较大，考虑到评分次数特别多的关键词对其他关键词的影响，引入 logistic 函数，对关键词评分次数进行非线性的映射，得到用户对关键词的兴趣度。

Logistic 函数于 1844 年由皮埃尔 •弗朗索瓦 •韦吕勒提出，用于描述生物种群发展、人类认真学习过程等，是一种常见的 S 型函数，定义公式如下，

$$S(x)=\frac{1}{1+e^{-x}} \tag{11}$$

函数光滑连续且严格单调，因变量在开始阶段随着变量增长而缓慢增长，发展期时极速增长，增长到一定程度之后，又增长减缓，其值无限趋近于 1。

用户对项目的兴趣度随着评分的次数增长呈非线性变化，在项目非独占的情况下，评分次数越高，就表示用户对该项目越感兴趣。因此，本文采用 logistic 函数模拟用户 u_i 与评分之间的关系，进而可以得到用户对项目的感兴趣程度，计算方法如式(12)所示。

$$H_{i,j}=\frac{1}{1+e^{-(R_{ij}-\overline{R}_i)}} \tag{12}$$

其中：$H_{i,j}$ 为用户 u_i 对项目 I_j 的兴趣度，取值为（0,1），随着评分次数增加而单调非线性递增，R_{ij} 为用户 u_i 对项目 I_j 的评分次数，$\overline{R}_i$ 为用户对所有项目的平均评分次数。

由式(12)可知，用户 u_i 对 $I=\{I_1,I_2,\cdots,I_n\}$ 中所有项目的兴趣度值构成了用户 u_i 兴趣度向量 $H_i=(H_{i,1},H_{i,2},\cdots,H_{i,m})$。在得到用户-项目兴趣度之后，需要在类簇中寻找与目标用户相似的 Top-N 邻居，计算方法如式(12)所示。

$$S(u_a,u_b)=\frac{\sum_{s_i\in S}[(H_{u_a,i}-\overline{H}_{u_a})(H_{u_b,i}-\overline{H}_{u_b})]}{\sqrt{\sum_{s_i\in S}(H_{u_a,i}-\overline{H}_{u_a})^2}\sqrt{\sum_{s_i\in S}(H_{u_b,i}-\overline{H}_{u_b})^2}} \tag{13}$$

其中：S 为用户 u_a 和 u_b 之间共同的感兴趣项目集合，$H_{u_a,i}$ 和 $H_{u_b,i}$ 分别表示用户 u_a 和 u_b 对项目 I_i 的兴趣度，$\overline{H}_{u_a}$ 和 $\overline{H}_{u_b}$ 分别表示用户 u_a 和 u_b 对集合 I 中项目的平均兴趣度。

2.3 预测与推荐

经过上节计算得到与目标用户相似的 Top-N 个用户，组成最近邻居集 N_{u_a}，根据式(13)预测目标用户对项目的偏好度，

$$P(u_a,t_q)=\overline{H}_a+\frac{\sum_{u_b\in N_{u_a}}[S_{u_au_b}\times(H_{b,q}-\overline{H}_b)]}{\sum_{u_b\in N_{u_a}}S_{u_au_b}} \tag{14}$$

其中：$\overline{H}_a$ 为用户 u_a 评分过的所有关键词的平均兴趣度，$H_{b,q}$ 为用户 u_b 对项目 I_q 的兴趣度。

得到目标用户对为评分过项目的评分预测值之后，将预测值最高的前 N 推荐给目标用户。

3 仿真结果分析

3.1 数据处理

仿真采用的是 GroupLens 项目研究组提供的 MovieLens 数据集[17]对算法进行评估。数据集中包含 943 个用户信息，1682 个项目信息，以及 100000 条用户对项目的评分信息，每个用户至少有 20 条评分记录，评分范围为 1(非常差)～5(非常好)。实验从数据集中随机选取 5 组数据且数据之间不相交，采用交叉验证法对算法进行评估验证。实验之前需要对数据集进行划分。本文实验将数据集的 80%作为训练集，剩余的 20%作为测试集。用户-项目评分数据如图 2 所示，每列依次表示：用户 ID | 项目 ID | 评分 | 时间戳，项目属性如图 3 所示，0 表示不具有该关键词属性，1 则表示项目具有关键词属性。关键词为 Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western |。

```
119 392 4   886176814
167 486 4   892738452
299 144 4   877881320
291 118 2   874833878
308 1   4   887736532
95  546 2   879196566
38  95  5   892430094
```

图 2 用户-项目评分数据样式

```
)|0|0|0|0|0|1|0|1|0|0|0|0|1|0|0|1|0|0
+(1998)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0
1998)|0|1|0|0|0|0|0|0|0|0|0|0|0|0|1|1|0|0
+(1998)|0|1|0|0|0|0|0|0|1|0|0|0|0|0|0|1|0|0
+(1998)|0|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0|0
```

图 3 项目属性数据样式

3.2 度量方法

本实验采用平均绝对误差(Mean Absolute Error，*MAE*)和准确率两种评价指标来度量算法的推荐质量。*MAE* 是一种常用的用于衡量统计的准确性和比较的度量方法，能够准确地反应推荐质量的好坏。它用来衡量预测的用户评分与实际用户评分之间的误差，*MAE* 值越小，推荐准确度越高，假设系统预测的用户兴趣集合为 $(p_1,p_2,\cdots p_n)$，其实际兴趣集合为 $(q_1,q_2,\cdots q_n)$ 计算方法如下：

$$MAE = \frac{\sum_{i=1}^{n} | p_i - q_i |}{n} \tag{15}$$

准确率指的是推荐系统为目标用户推荐其感兴趣项目的概率，具体计算方法如下：

$$Precision = \frac{C}{N} \tag{16}$$

其中， N 为推荐系统产生的推荐总数， C 为系统产生的正确的推荐数目。

3.3 仿真结果及分析

实验主要分为两个部分：

实验 1　分析算法中主要参数对推荐效果的影响，主要从关键词数目方面分析。

实验 2　为了验证本文算法的有效性进行对比实验，通过相同参数环境下，比较本文算法与现存算法的 *MAE* 值，准确率等。

实验 1

本文算法主要根据目标用户对不同关键词的兴趣度用户推荐，为了考察关键词数目对于推荐结果是否有影响，设计实验考察不同关键词数目对 *MAE* 的影响，实验结果如图 4 所示。

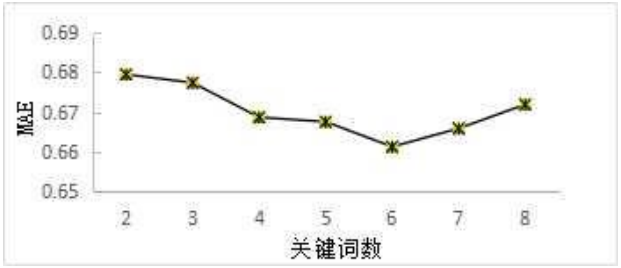

图 4　关键词数对 *MAE* 值的影响

由图 4 可以看出，当关键词数目取 6 时， *MAE* 值最小，算法效果最好。当关键词数目较少时，很难计算用户之间的相似度，进而影响算法推荐效果；当数目过多时，又会增加矩阵的稀疏度，导致相似度计算结果受影响，使得误差升高。

实验 2

上节中给出了关键词数目对算法的影响，因此，本是要的算法取关键词数为 6，进行实验，将本文算法与现有算法从 *MAE* 值和准确率两个方面进行比较，算法对 *MAE* 值的影响如图 5 所示，对准确率的影响如图 6 所示，运行效率如图 7 所示。

从图 5 可以看出本文算法随着邻居数目的增加， *MAE* 值不断减小，且逐渐趋于平稳。邻居数目为 40 时，本文算法的 *MAE* 值约为 0.643，而基于 logistic 聚类的算法 *MAE* 值约为 0.676，本文算法始终优于传统算法。

由图 6 可以看出，本文算法随着邻居数目的增加，准确率不断增高，且逐渐趋于平稳。邻居数目为 20 时，本文算法准确率最高，约为 0.2487，而文献[15]算法准确率约为 0.2442，且本文算法始终优于文献[15]算法。

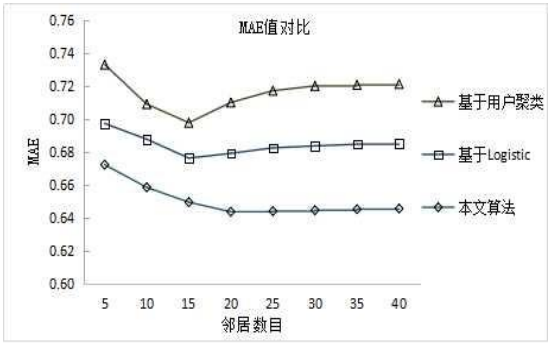

图 5　不同算法的 *MAE* 值

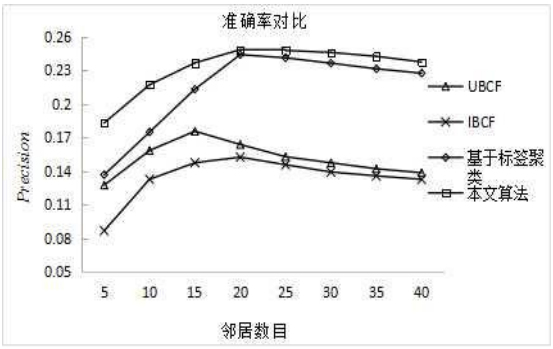

图 6　不同算法的准确率比较

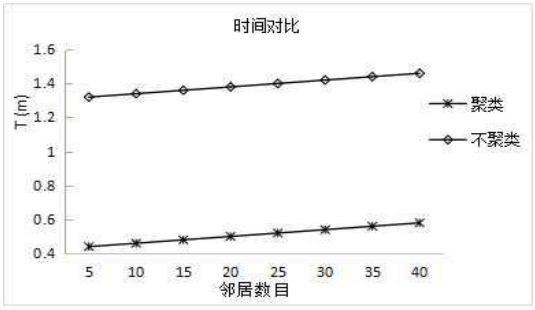

图 7　采用不同方法产生推荐所需时间

图 7 是算法在聚类与不聚类两种情况下产生推荐的时间，从图中可以看出，随着邻居数目的增加，两种情况的时间都在增加，然而采用聚类的时候算法运行时间远远小于不采用聚类的情况，推荐效率明显提高。

4　结束语

本文引入 logistic 函数模拟用户对项目的评分次数的非线性关系，用于计算用户对项目关键词的兴趣度，形成用户-关键词兴趣度矩阵，从而构建用户-项目兴趣度矩阵，在新的矩阵基础上进行用户聚类，然后在目标用户所在的类簇中寻找最近邻居，缩小搜索范围，提升了算法的效率。经实验表明，该算法有效的利用了用户对项目关键词的兴趣度，能够有效的提取出用户的最近邻居，提高了算法的准确率。然而在本文算法中未详细考虑随着时间的变化，用户的兴趣也会发生变化，下一步将考虑时间戳的影响，结合遗忘机制来对用户进行推荐，进一步提高推荐进度。

参考文献：

[1] 项亮. 推荐系统实践 [M]. 北京: 人民邮电出版社, 2012. (Xiang Liang.

Recommended system practice [M]. Beijing: People's Posts and Telecommuni-cations Press, 2012.)

[2] Herlocker L, Konstan A, BorchersS A, *et al.* An algorithmic framework for performing collabora-tive filtering [C]// Proc of, International ACM SIGIR Conference on Research and Development in Information Retrieval. 1999: 230-237.

[3] Dehghani Z, Reza S, Salwah S, *et al.* A systematic review of scholar context-aware recommender systems [J]. Expert Syst. Appl. 2015 (42): 1743.

[4] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions [J]. IEEE Trans on Knowledge & Data Engineering, 2005, 17 (6): 734-749

[5] Mnih A, Salakhutdinov R. Probabilistic matrix factorization [C]// Advances in Neural Information Processing Systems. 2007: 1257.

[6] Paterek A. Improving regularized singular value decomposition for collaborative filtering [C]// Proc of KDD Cup Workshop at SIGKDD&the 13th ACM Int Conf on Knowledge Discovery and Data Mining. 2007: 39.

[7] You Haipeng, Li Hui, Wang Yunmin, *et al.* An improved collaborative filtering recommendation algorithm combining item clustering and slope one scheme [C]. Lecture Notes in Engineering & Computer Science, vol 2215. 2015: 313-316.

[8] 高凤荣，邢春晓，杜小勇，等. 基于矩阵聚类的协作过滤算法 [J]. 华中科技大学学报：自然科学版, 2005, 33 (S1): 257-260. (Gao Fengrong, Xing Chunxiao, Du Xiaoyong, Wang Shan. A collaborative filtering algorithm based on matrix clustering [J]. Journal of Huazhong University of Science and Technology (Natural Science Edition) , 2005, 33 (S1): 257-260.)

[9] 兰 艳，曹芳芳. 面向电影推荐的时间加权协同过滤算法的研究 [J]. 计算机科学, 2017, 44 (4): 295-301. (Lan Yan, Cao Fangfang. A temporal weighted collaborative filtering algorithm for movie recommendation [J]. Computer Science, 2017, 44 (4): 295-301.)

[10] 范波，程久军. 用户间多相似度协同过滤推荐算法 [J]. 计算机科学, 2012, 39 (1): 23-26. (Fan Bo, Cheng Jiujun. Among multiple users similarity collaborative filtering algorithm [J]. Computer Science, 2012, 01: 23-26.)

[11] Zhao Z D, Shang M S. User-based collaborative filtering recommendation algorithms on Hadoop [C]// Proc of the 3rd International Conference on Knowledge Discovery and Data Mining. 2010: 478-481.

[12] Herlocker J L. Evaluating collaborative filtering recommender systems [J]. Acm Trans on Information Systems, 2004, 22 (1): 5-53.

[13] 黄震华，张佳雯，田春岐，等. 基于排序学习的推荐算法研究综述 [J]. 软件学报, 2016, 27 (3): 691-713. (Huang Zhenhua, Zhang Jiawen, Tian Chunqi, *et al.* A survey of recommendation algorithms based on ranking learning. [J]. Software Journal, 2016, 27 (3): 691-713)

[14] 张松，张琳，王汝传. 基于用户限制聚类的协同过滤推荐算法 [J]. 南京邮电大学学报：自然科学版, 2017. 37 (3): 93-99. (Zhang Song, Zhang Lin, Wang Ruchuan. Collaborative filtering recommendation algorithm based on user restricted clustering [J]. Nanjing University of Posts and Telecommunications: Natural Science Edition, 2017, 37 (3): 93-99.)

[15] 朱东郡，李敬兆，谭大禹，等. 基于标签聚类和兴趣划分的协同过滤推荐算法 [J]. 计算机工程, 2017, 43 (11): 146-151. (Zhu Dongjun, Li Jingzhao, Tan Dayu, *et al.* Collaborative filtering recommendation algorithm based on tag clustering and interest partition [J]. Computer Engineering, 2017. 43 (14): 146.)

[16] Forsati R, Barjasteh I, Masrour F, *et al.* PushTrust: an efficient recommendation algorithm by leveraging trust and distrust relations [C]// Proc of Conference on Recommender Systems. 2015: 51-58

[17] MovieLens_100K [DB/OL]. https://grouplens. org/datasets/movielens/.